

A model selection approach for multiple sequence segmentation and dimensionality reduction

Bruno M. de Castro and Florencia Leonardi

January 9, 2015

Abstract

In this paper we consider the problem of segmenting n aligned random sequences of equal length m into a finite number of independent blocks. We propose to use a penalized maximum likelihood criterion to infer simultaneously the number of points of independence as well as the position of each one of these points. We show how to compute the estimator efficiently by means of a dynamic programming algorithm with time complexity $O(m^2n)$. We also propose another algorithm, called *hierarchical algorithm*, that provides an approximation to the estimator when the sample size increases and runs in time $O(mn)$. Our main theoretical result is the proof of almost sure consistency of the estimator and the convergence of the hierarchical algorithm when the sample size n grows to infinity. We illustrate the convergence of these algorithms through some simulation examples and we apply the method to real protein sequence alignments of Ebola Virus.

1 Introduction

The problem of multiple sequence segmentation and dimensionality reduction appears in many application areas, including the analysis of multiple alignments of DNA/RNA and Amino Acid (AA) sequences. These datasets are characterized by the alignment of sequences belonging to the same group, as genes, protein families, etc. In general the length of these alignments is big and direct multivariate analysis is not possible. On the other hand, the identification of relevant patterns in these datasets is crucial for many biological problems, as for example the identification of specific patterns associated to some phenotypic variables. An incomplete list of methods for sequence segmentation and pattern identification of biological sequences include [3, 8, 2].

In this paper we consider the problem of segmenting n aligned random sequences of equal length m into a finite number of independent blocks. To be more precise, let $\vec{X} = (X_1, \dots, X_N)$ be a random vector with probability distribution \mathbb{P} taking values in $A_1 \times \dots \times A_N$, where A_i is a finite alphabet for all $i = 1, \dots, N$. We say $j = i + \frac{1}{2}$, with $i \in \{1, 2, \dots, N-1\}$, is a point of independence for \vec{X} if the random vectors (X_1, \dots, X_j)

and (X_{j+1}, \dots, X_N) are independent. We are interested in inferring the maximal set of points of independence of a random vector, when the number of such points is unknown. This is a model selection problem because the different parameter sets of the probability distributions belong to vector spaces with different dimensions. We propose a penalized maximum likelihood criterion to infer simultaneously the number of points of independence and their positions. Our main theoretical result is the proof of the almost sure convergence of the estimator to the true set of points of independence when the sample size n increases.

From the computational point of view we show how to compute the estimator by means of a dynamic programming algorithm that runs in time $O(m^2n)$. When the size of the sequences is large this can be time consuming. In these situations we propose a suboptimal algorithm running in time $O(mn)$ that also converges almost surely to the set of points of independence when the sample size n increases.

To our knowledge this is the first model selection approach for multiple sequence segmentation based on the assumption of independence between segments. The change point detection problem is a well studied phenomenon in the area of time series analysis (see for example [7] and references therein) but in general the setting and the goals are different from ours.

In the next section we present the main definitions and state the main theoretical result. In Section 3 we describe the algorithm to compute exactly the estimator and we present the *hierarchical algorithm* that provides an efficient but suboptimal solution. In Section 4 we prove the theoretical results and in Sections 5 and 6 we present the simulations and results on real data, respectively.

2 Likelihood function and model selection

Given two integers $r \leq s$ denote by $r:s$ the integer interval $r, r+1, \dots, s$. Denote by $J_{r:s}$ the set $\{i + \frac{1}{2} : i = r, 1, \dots, s-1\}$. We say $U_{r:s} \subset J_{r:s}$ is a *maximal* set of points of independence for the interval $r:s$ if no $v \in J_{r:s} \setminus U_{r:s}$ is a point of independence for \vec{X} . For each random vector \vec{X} and each interval $r:s$ there is only one maximal set of points of independence; from now on this special set will be denoted by $U_{r:s}^*$. In the special case $r = 1, s = N$ we will simply write U , J and U^* .

Without loss of generality we will also suppose that the set $U_{r:s} \subset J_{r:s}$ is ordered; in this case $U_{r:s} = (u_1, \dots, u_k)$ with $u_i < u_j$ if $i < j$. From $U_{r:s}$ it is possible to obtain the set of blocks of independent variables as the set $B(U_{r:s}) = \{I_1, I_2, \dots, I_{k+1}\}$ of integer intervals given by

$$I_1 = r : (u_1 - \frac{1}{2}), \quad I_i = (u_{i-1} + \frac{1}{2}) : (u_i - \frac{1}{2}), \quad i = 2, \dots, k$$

and

$$I_{k+1} = (u_k + \frac{1}{2}) : s.$$

Given an integer interval $I = r:s$ denote by A^I the set of finite sequences $A_r \times \dots \times A_s$ and by $|A^I| = \prod_{i=r}^s |A_i|$, with $|A_i|$ the cardinal of the set A_i .

In this way, given a vector of observations $\mathbf{x} = (x_1, \dots, x_N)$, the likelihood function for the set U can be written as

$$L(U; \mathbf{x}) = \prod_{I \in B(U)} \mathbb{P}(X_j = x_j : j \in I).$$

Assume we observe an i.i.d sample $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}$ of \vec{X} , denoted by \mathbf{X} . Then we have

$$L(U; \mathbf{X}) = \prod_{i=1}^n \prod_{I \in B(U)} \mathbb{P}(X_j = x_j^{(i)} : j \in I).$$

Denote by $X_{r:s}$ the sequence X_r, \dots, X_s and by $\mathbf{X}_{r:s}$ the i.i.d sample $\mathbf{x}_{r:s}^{(1)}, \dots, \mathbf{x}_{r:s}^{(n)}$. Given a finite sequence $a_{r:s} \in A^{r:s}$ define

$$N(a_{r:s}) = \sum_{i=1}^n \mathbf{1}\{x_{r:s}^{(i)} = a_{r:s}\}.$$

Then $L(U; \mathbf{X})$ can be rewritten as

$$L(U; \mathbf{X}) = \prod_{I \in B(U)} \prod_{a_I \in A^I} \mathbb{P}(X_I = a_I)^{N(a_I)}. \quad (2.1)$$

To proceed we need a model for the block I and some estimators for the probabilities $\mathbb{P}(X_I = a_I)$. Assume that for each block $I = r:s \subset 1:N$ we have a model assignment $\mathcal{M}(I)$ and a penalization function $p(\mathbf{X}_I)$ satisfying

(A1) For any $i \in r:(s-1)$ we have

$$p(\mathbf{X}_{r:s}) > p(\mathbf{X}_{r:i}) + p(\mathbf{X}_{(i+1):s}).$$

We will assume that in model $\mathcal{M}(I)$ we have maximum likelihood estimators for the probabilities $\mathbb{P}(a_I)$; that is some values $\hat{\mathbb{P}}(a_I)$ maximizing (2.1). Then we can plug-in these estimators in (2.1) obtaining

$$\hat{L}(U; \mathbf{X}) = \prod_{I \in B(U)} \prod_{a_I \in A^I} \hat{\mathbb{P}}(a_I)^{N(a_I)}.$$

Remark 2.1. As an example we could consider for any block I the multivariate distribution over A^I as possible model assignment $\mathcal{M}(I)$. For this class of models we have that the maximum likelihood estimators are given by

$$\hat{\mathbb{P}}(a_I) = \frac{N(a_I)}{n}, \quad a_I \in A^I.$$

In this case, for example, we could take as penalizing function $p(\mathbf{X}_I) = c(|A^I| - 1)$, the number of parameters in the multivariate distribution over A^I (in case A^I is known),

multiplied by a positive constant c . In the case A^I is unknown a priori, it could be appropriate to use an empirical version of $p(I)$; namely $\hat{p}(\mathbf{X}_I) = c(\prod_{i \in I} \|\mathbf{X}_i\|_0 - 1)$, where $\|\mathbf{X}_i\|_0 = \sum_{a_i \in A_i} \mathbf{1}\{N(a_i) > 0\}$ and c is a positive constant. In some cases other model assignments or penalizing functions could be more efficient or parsimonious, mainly in the case of large blocks I . One example is to consider a Markov chain of some appropriate order.

In the sequel it will be easier to work with the logarithm of the estimated likelihood function, for this reason we define

$$\hat{\ell}(U; \mathbf{X}) = \sum_{I \in B(U)} Q(I, \mathbf{X}),$$

where

$$Q(I, \mathbf{X}) = \sum_{a_I \in A^I} N(a_I) \log \hat{\mathbb{P}}(a_I).$$

Now we introduce the model selection criterion based on the maximization of the penalized log-likelihood.

Definition 2.2. Given a sample \mathbf{X} define

$$\hat{U}(\mathbf{X}) = \arg \max_{U \subset J, U \text{ ordered}} \{ \hat{\ell}(U, \mathbf{X}) - \text{pen}(U, n) \}, \quad (2.2)$$

where

$$\text{pen}(U, n) = \sum_{I \in B(U)} p(\mathbf{X}_I) \log(n).$$

The first theoretical result of this article shows that when n grows to infinity, the estimator (2.2) is equal to the true set U^* eventually almost surely. In other words, with probability one there exists n_0 (depending on the entire sample $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}$) such that for all $n \geq n_0$ we have $\hat{U}(\mathbf{X}) = U^*$.

Theorem 2.3. *The estimator (2.2) is strongly consistent; that is*

$$\hat{U}(\mathbf{X}) = U^*$$

eventually almost surely when $n \rightarrow \infty$.

The proof of Theorem 2.3 is postponed to Section 4.

3 Computation of the independence set estimator

In this section we show how to compute efficiently the penalized maximum likelihood estimator given in Definition 2.2. The first subsection, mostly inspired in [9], presents a dynamic programming algorithm that computes exactly the optimal argument of (2.2), at the cost of a quadratic computing time in terms of the length of the sequences (see also [1] and references therein). In the second subsection we propose a *divide and conquer* approximation for the optimum in (2.2), at a more efficient computing time. We show that this second algorithm also retrieves the true set of points of independence with probability one when the sample size grows.

3.1 Dynamic programming algorithm

Let $F_{k+1}(N)$ denote the maximum value of the function in (2.2) corresponding to a k -dimensional vector U for the sample \mathbf{X} ; that is

$$F_{k+1}(N) = \max_{U, |U|=k} \{ \hat{\ell}(U, \mathbf{X}) - \text{pen}(U, n) \}.$$

It is easy to see that the optimal k -dimensional vector U leading to $F_{k+1}(N)$ consist of $k - 1$ independence points over $1:i$ and a single block $(i + 1):N$, where $i + 1/2$ is the rightmost point of independence. Moreover, the k blocks over $1:i$ must maximize the function (2.2) for the sample $\mathbf{X}_{1:i}$, attaining $F_k(i)$. In this way, the dynamic programming recursion is

$$F_1(N) = \tilde{Q}(1:N, \mathbf{X}), \quad F_{k+1}(N) = \max_{i=k, \dots, N-1} \{ F_k(i) + \tilde{Q}((i + 1):N, \mathbf{X}) \},$$

where \tilde{Q} is given by

$$\tilde{Q}(I, \mathbf{X}) = Q(I, \mathbf{X}) - p(\mathbf{X}_I) \log(n).$$

The estimator $\hat{U}(\mathbf{X})$ in Definition 2.2 is computed by tabulating $F_1(i)$ for all i up to N , and then by computing $F_2(i)$ for all i and so on up to $F_N(N)$. The optimal value of k is obtained by the equation

$$\hat{k} = \arg \max_{k=1, \dots, N} \{ F_k(N) \} - 1.$$

and the vector $\hat{U}(\mathbf{X}) = (\hat{u}_1, \dots, \hat{u}_{\hat{k}})$ is given by

$$\begin{aligned} \hat{u}_{\hat{k}} &= \arg \max_{i=\hat{k}, \dots, N-1} \{ F_{\hat{k}}(i) + \tilde{Q}((i + 1):N, \mathbf{X}) \} + \frac{1}{2}, \\ \hat{u}_i &= \arg \max_{\ell=i, \dots, \hat{u}_{i+1}} \{ F_i(\ell) + \tilde{Q}((\ell + 1):N, \mathbf{X}) \} + \frac{1}{2}, \quad i = 1, \dots, \hat{k} - 1. \end{aligned}$$

3.2 Hierarchical Algorithm

Here we present an efficient *divide and conquer* algorithm to compute the estimator given by Definition 2.2 with computational cost $O(nm)$.

Let $I = r:s$ be an integer interval (with the convention that $I = \emptyset$ if $s < r$). Define

$$h(I, \mathbf{X}) = \arg \max_{i \in I} \{ \tilde{Q}(r:(i - 1), \mathbf{X}) + \tilde{Q}(i:s, \mathbf{X}) \} - r - \frac{1}{2}, \quad (3.1)$$

where also by convention we have $\tilde{Q}(\emptyset, \mathbf{X}) = 0$.

The idea of this algorithm is to compute the best point of independence for the interval I (if there is one point in the interval leading to a maximum of the penalized likelihood or a point outside I otherwise) and then to iterate this criterion on both

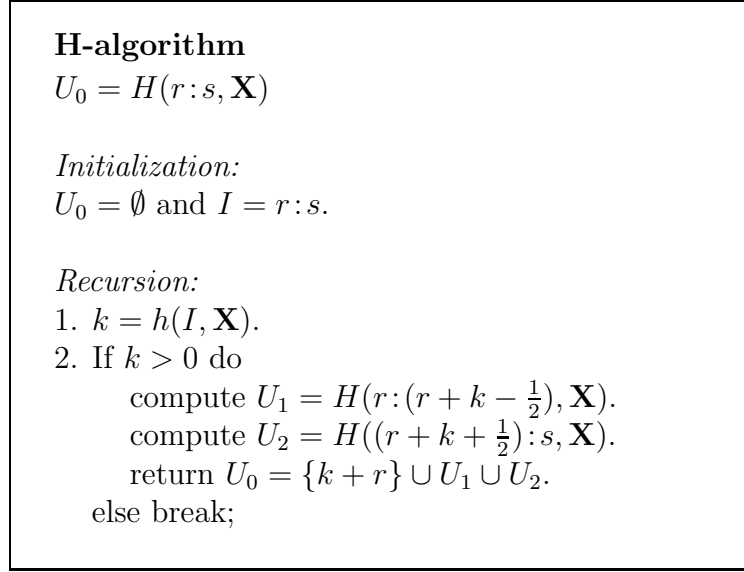


Figure 1: H -algorithm.

segments separated by this point, until no more points are detected. A summary of the Hierarchical algorithm is given in Figure 1.

We show the almost sure convergence of the hierarchical algorithm in the following theorem.

Theorem 3.1. $H(1:N, \mathbf{X}) = U^*$ eventually almost surely as $n \rightarrow \infty$.

The proof of this theorem is given in the following section.

4 Proofs

Let $I \subset 1:N$ be an integer interval. Given two probability distributions P_1 and P_2 over A^I let $D(P_1\|P_2)$ denote the *Kullback-Leibler* divergence between P_1 and P_2 ; that is

$$D(P_1\|P_2) = \sum_{a_I: P_1(a_I) > 0} P_1(a_I) \log \left(\frac{P_1(a_I)}{P_2(a_I)} \right).$$

A well-know property about the Kullback-Leibler divergence between two probability distributions states that $D = 0$ if and only if $P_1 = P_2$.

Lemma 4.1. *For any block $I \subset 1:N$ and any $\alpha > 0$ we have that*

$$D(\widehat{\mathbb{P}}\|\mathbb{P}) < \frac{\alpha \log(n)}{n}$$

eventually almost surely as $n \rightarrow \infty$, where $\widehat{\mathbb{P}}$ and \mathbb{P} represent the marginal distributions over the interval I .

Proof. Define, for a fixed $a_I \in A^I$ with $\mathbb{P}(a_I) > 0$, the random variables

$$Y_i(a_I) = \mathbf{1}\{\mathbf{x}_I^{(i)} = a_I\} - \mathbb{P}(a_I), \quad i = 1, 2, \dots, n$$

and

$$Z_n(a_I) = \sum_{i=1}^n Y_i(a_I) = N(a_I) - n\mathbb{P}(a_I).$$

The variables $\{Y_i(a_I) : i = 1, 2, \dots, n\}$ are independent and identically distributed, with $\mathbb{E}(Y_i(a_I)) = 0$ and $\mathbb{E}(Y_i(a_I)^2) = \mathbb{P}(a_I)(1 - \mathbb{P}(a_I))$. Then, by the Law of the Iterated Logarithm (cf. Theorem 3.52 in [4]) we have that for any $\epsilon > 0$

$$|Z_n(a_I)| < (1 + \epsilon) \mathbb{P}(a_I)(1 - \mathbb{P}(a_I)) \sqrt{2n \log \log n} \quad (4.1)$$

eventually almost surely as $n \rightarrow \infty$. Dividing both sides of the last inequality by $n\sqrt{\mathbb{P}(a_I)}$ we obtain that for any $\delta > 0$

$$\frac{|Z_n(a_I)/n|}{\sqrt{\mathbb{P}(a_I)}} = \frac{|\hat{\mathbb{P}}(a_I) - \mathbb{P}(a_I)|}{\sqrt{\mathbb{P}(a_I)}} < \sqrt{\frac{\delta \log(n)}{n}} \quad (4.2)$$

eventually almost surely as $n \rightarrow \infty$. Now by Lemma 6.3 in [5] and (4.2) we have that

$$\begin{aligned} D(\hat{\mathbb{P}} \parallel \mathbb{P}) &\leq \sum_{a_I: N(a_I) > 0} \frac{|\hat{\mathbb{P}}(a_I) - \mathbb{P}(a_I)|^2}{\mathbb{P}(a_I)} \\ &\leq |A^I| \max_{a_I: \mathbb{P}(a_I) > 0} \frac{|\hat{\mathbb{P}}(a_I) - \mathbb{P}(a_I)|^2}{\mathbb{P}(a_I)} \\ &\leq \frac{\delta \log(n)}{|A^I|n} \end{aligned}$$

eventually almost surely as $n \rightarrow \infty$ and this implies the result for $|A^I|$ finite. \square

For any $j \in I = r:s$ denote by $\tilde{\mathbb{P}}_j$ the probability distribution given by

$$\tilde{\mathbb{P}}_j(a_I) = \mathbb{P}(a_{r:j})\mathbb{P}(a_{(j+1):s}), \quad a_I \in A^I.$$

Lemma 4.2. *Let $I = r:s \subset 1:N$ and suppose $I \subset I^*$, with $I^* \in B(U^*)$; that is, there is no point of independence in the range of I . Then there exists a constant $\delta > 0$ such that for any $i \in I, i \neq s$ we have*

$$Q(I; \mathbf{X}) - Q(r:i; \mathbf{X}) - Q((i+1):s; \mathbf{X}) > \delta n$$

eventually almost surely when $n \rightarrow \infty$.

Proof. Note that for any $a_r^i \in A^{r:i}$ we have $N(a_r^i) = \sum_{a_{i+1}^s} N(a_r^i a_{i+1}^s)$ and analogously for any $a_{i+1}^s \in A^{(j+1):s}$, $N(a_{i+1}^s) = \sum_{a_r^i} N(a_r^i a_{i+1}^s)$. Then

$$Q(I; \mathbf{X}) - Q(r:i; \mathbf{X}) - Q((i+1):s; \mathbf{X}) = \sum_{a_r^s \in A^I} N(a_r^s) \log \left(\frac{\widehat{\mathbb{P}}(a_r^s)}{\widehat{\mathbb{P}}(a_r^i) \widehat{\mathbb{P}}(a_{i+1}^s)} \right)$$

Dividing by n and taking limit when $n \rightarrow \infty$ we have that the expression above converges almost surely to

$$\sum_{a_r^s \in A^I} \mathbb{P}(a_r^s) \log \left(\frac{\mathbb{P}(a_r^s)}{\mathbb{P}(a_r^i) \mathbb{P}(a_{i+1}^s)} \right) = D(\mathbb{P} \parallel \widetilde{\mathbb{P}}_j) > 0.$$

The last inequality follows because $i + \frac{1}{2}$ is not a point of independence for \vec{X} . To conclude the proof, it is enough to take

$$\delta = \min_{j \in I, j \neq s} D(\mathbb{P} \parallel \widetilde{\mathbb{P}}_j) / 2. \quad \square$$

Lemma 4.3. *Let $I = r:s \subset 1:N$ and suppose there exists $i \in I$ such that $i + \frac{1}{2} \in U^*$; that is, there is a point of independence in the range of I . Then for any $\alpha > 0$ we have*

$$Q(I; \mathbf{X}) - Q(r:i; \mathbf{X}) - Q((i+1):s; \mathbf{X}) < \alpha \log(n)$$

eventually almost surely when $n \rightarrow \infty$.

Proof. As in the proof of Lemma 4.2 we can write

$$Q(I; \mathbf{X}) - Q(r:i; \mathbf{X}) - Q((i+1):s; \mathbf{X}) = \sum_{a_r^s \in A^I} N(a_r^s) \log \left[\frac{\widehat{\mathbb{P}}(a_r^s)}{\widehat{\mathbb{P}}(a_r^i) \widehat{\mathbb{P}}(a_{i+1}^s)} \right].$$

As $\widehat{\mathbb{P}}(\cdot)$ is the maximum likelihood estimator of $\mathbb{P}(\cdot)$ and $i + \frac{1}{2}$ is a point of independence we have that

$$\begin{aligned} Q(r:i; \mathbf{X}) + Q((i+1):s; \mathbf{X}) &\geq \sum_{a_r^s \in A^I} N(a_r^s) \log [\mathbb{P}(a_r^i) \mathbb{P}(a_{i+1}^s)] \\ &= \sum_{a_r^s \in A^I} N(a_r^s) \log [\mathbb{P}(a_r^s)] \end{aligned}$$

Then by combining this last inequality and Lemma 4.1 we have that

$$\begin{aligned} Q(I; \mathbf{X}) - Q(r:i; \mathbf{X}) - Q((i+1):s; \mathbf{X}) &\leq \sum_{a_r^s \in A^I} N(a_r^s) \log \left[\frac{\widehat{\mathbb{P}}(a_r^s)}{\mathbb{P}(a_r^s)} \right] \\ &< \alpha \log(n) \end{aligned}$$

eventually almost surely as $n \rightarrow \infty$. \square

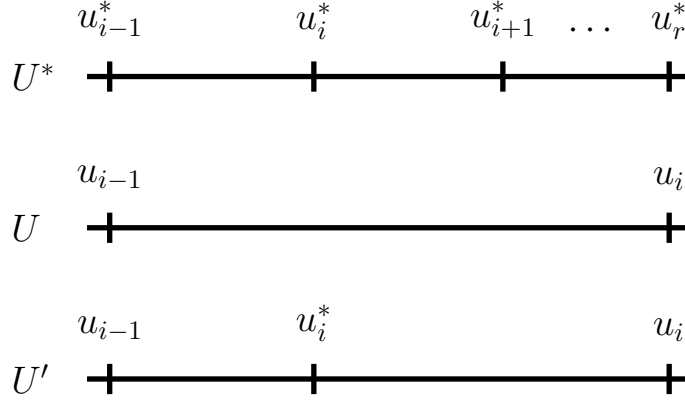


Figure 2: Case (a): $U \subset U^*$.

Proof of Theorem 2.3. We will show that if $U \subset J$ (ordered) is such that $U \neq U^*$ then there exists a modification U' of U such that

$$\hat{\ell}(U'; \mathbf{X}) - \text{pen}(U', n) > \hat{\ell}(U; \mathbf{X}) - \text{pen}(U, n) \quad (4.3)$$

eventually almost surely as $n \rightarrow \infty$. As the number of possible such modifications of U leading to U^* is finite this implies the result. Consider the following cases

- (a) $U \subset U^*$
- (b) $U \not\subset U^*$.

First we will prove (4.3) for case (a). Note that in this case it is enough to show it in the case $U^* \neq \emptyset$. Assume $U^* = (u_1^*, \dots, u_K^*)$, with $K \geq 1$, $U = (u_1, \dots, u_k)$ and let i be the first index such that $u_i^* \notin U$. Define $U' = U \cup \{u_i^*\} = (u_1, \dots, u_{i-1}, u_i^*, u_{i+1}, \dots, u_k)$ (see Figure 2).

Note that $B(U')$ has the two blocks $I_1 = (u_{i-1} + \frac{1}{2}) : (u_i^* - \frac{1}{2})$ and $I_2 = (u_i^* + \frac{1}{2}) : (u_i - \frac{1}{2})$ replacing the single block $I = (u_{i-1} + \frac{1}{2}) : (u_i - \frac{1}{2})$ in $B(U)$. Then by Lemma 4.3 we have that for any $\alpha > 0$

$$\begin{aligned} \hat{\ell}(U'; \mathbf{X}) - \hat{\ell}(U; \mathbf{X}) &= Q(I_1, \mathbf{X}) + Q(I_2, \mathbf{X}) - Q(I, \mathbf{X}) \\ &> -\alpha \log(n) \end{aligned}$$

eventually almost surely as $n \rightarrow \infty$. On the other hand

$$\text{pen}(U', n) - \text{pen}(U, n) = c_1 \log(n),$$

where $c_1 = p(\mathbf{X}_{I_1}) + p(\mathbf{X}_{I_2}) - p(\mathbf{X}_I) < 0$. Therefore eventually almost surely as $n \rightarrow \infty$ we have that

$$\hat{\ell}(U'; \mathbf{X}) - \hat{\ell}(U; \mathbf{X}) > \text{pen}(U', n) - \text{pen}(U, n).$$

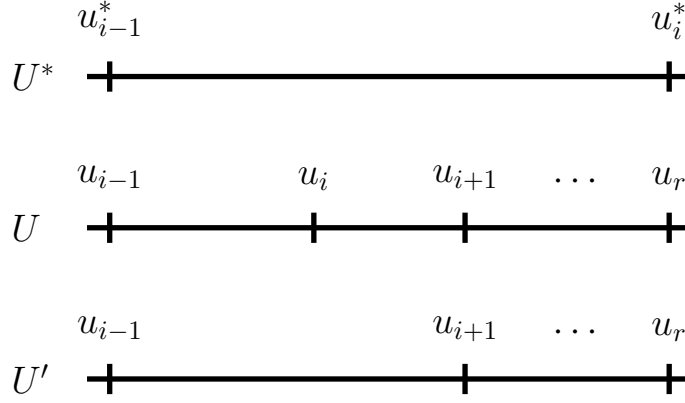


Figure 3: Case (b): $U \not\subset U^*$.

Finally we will consider case (b). As before suppose $U^* = (u_1^*, \dots, u_K^*)$, $U = (u_1, \dots, u_k)$ and let i be the first index such that $u_i \notin U^*$. Define $U' = U \setminus \{u_i\} = (u_1, \dots, u_{i-1}, u_{i+1}, \dots, u_k)$ (see Figure 3). In this case we have that $B(U')$ has the single block $I = (u_{i-1} + \frac{1}{2}) : (u_{i+1} - \frac{1}{2})$ replacing the two blocks $I_1 = (u_{i-1} + \frac{1}{2}) : (u_i - \frac{1}{2})$ and $I_2 = (u_i + \frac{1}{2}) : (u_{i+1} - \frac{1}{2})$ in $B(U)$. Then

$$\hat{\ell}(U'; \mathbf{X}) - \hat{\ell}(U; \mathbf{X}) = Q(I, \mathbf{X}) - Q(I_1, \mathbf{X}) - Q(I_2, \mathbf{X}).$$

By Lemma 4.2, as $u_i \notin U^*$ we have that there exists a constant $\delta > 0$ such that eventually almost surely

$$\hat{\ell}(U'; \mathbf{X}) - \hat{\ell}(U; \mathbf{X}) > \delta n.$$

On the other hand we have that

$$\text{pen}(U', n) - \text{pen}(U, n) = c_2 \log(n),$$

where $c_2 = p(\mathbf{X}_I) - p(\mathbf{X}_{I_1}) - p(\mathbf{X}_{I_2}) > 0$. Therefore eventually almost surely as $n \rightarrow \infty$ we have that

$$\hat{\ell}(U'; \mathbf{X}) - \hat{\ell}(U; \mathbf{X}) > \text{pen}(U', n) - \text{pen}(U, n).$$

This concludes the proof of Theorem 2.3. \square

Proof of Theorem 3.1. First consider the case $U^* = \emptyset$. By Lemma 4.2 we have that eventually almost surely as $n \rightarrow \infty$ the value of i maximizing (3.1) will be $i = r$, giving $h(1:N, \mathbf{X}) = -\frac{1}{2}$. Therefore $H(1:N, \mathbf{X}) = \emptyset = U^*$ eventually almost surely as $n \rightarrow \infty$. Now suppose there is a point of independence $u \in U^*$. We will prove that eventually almost surely $u \in H(1:N, \mathbf{X})$. As $u \in U^*$, for any integer $r < u$ and any integer $s > u$ we have by Lemma 4.3 that for any $\alpha > 0$

$$Q(r : (u - \frac{1}{2}), \mathbf{X}) + Q((u + \frac{1}{2}) : s, \mathbf{X}) - Q(r : s, \mathbf{X}) > -\alpha \log(n)$$

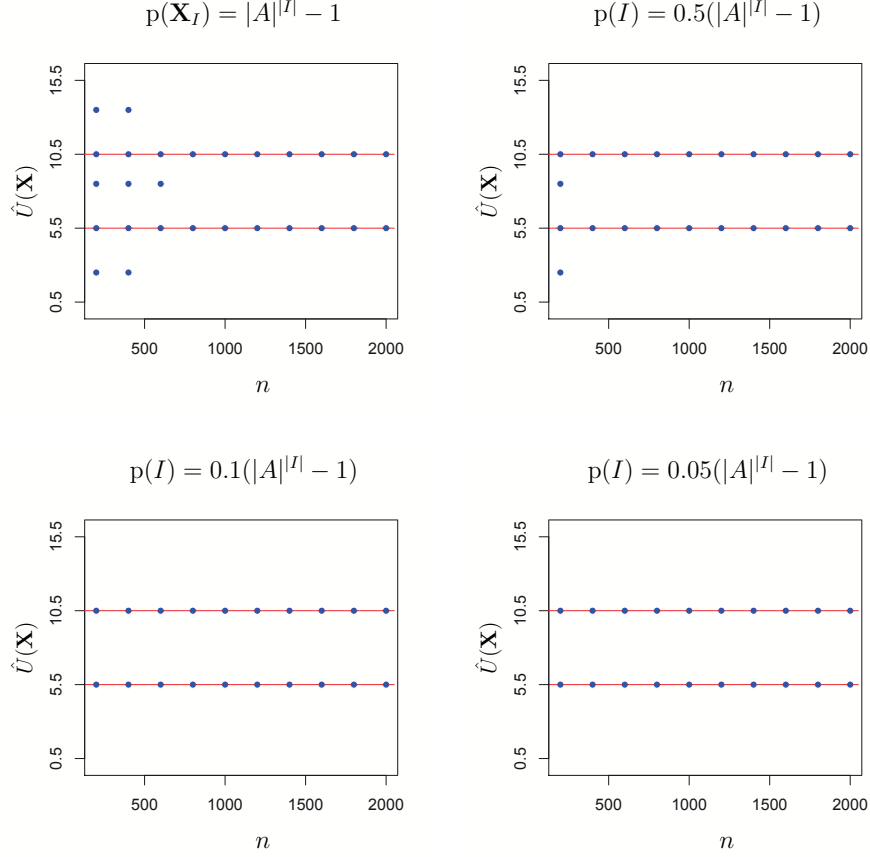


Figure 4: Dynamic programming estimator of the set of points of independence $U^* = \{5.5, 10.5\}$, with $|A| = 2$ and for different penalizing functions.

eventually almost surely as $n \rightarrow \infty$. Then if we take $\alpha < p(\mathbf{X}_{r:s}) - p(\mathbf{X}_{r:(u-\frac{1}{2})}) - p(\mathbf{X}_{u+\frac{1}{2}:s})$ we will have that eventually, as the interval length $s - r$ decreases, $h(r : s, \mathbf{X}) = u - r$, or equivalently $u \in H(r : s, \mathbf{X}) \subset H(1 : N, \mathbf{X})$. As in any iteration of the algorithm there is an interval that contains u and the length decreases, this will happen eventually almost surely as $n \rightarrow \infty$. \square

5 Simulations

We perform some simulations and test both the dynamic programming (exact estimator) as well as the hierarchical algorithm. First we considered the alphabet $A = \{1, 2\}$ and the transition matrices P_1 and P_2 given by

$$P_1 = \begin{pmatrix} \frac{1}{6} & \frac{5}{6} \\ \frac{5}{6} & \frac{1}{6} \end{pmatrix}, \quad P_2 = \begin{pmatrix} \frac{5}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{5}{6} \end{pmatrix}.$$

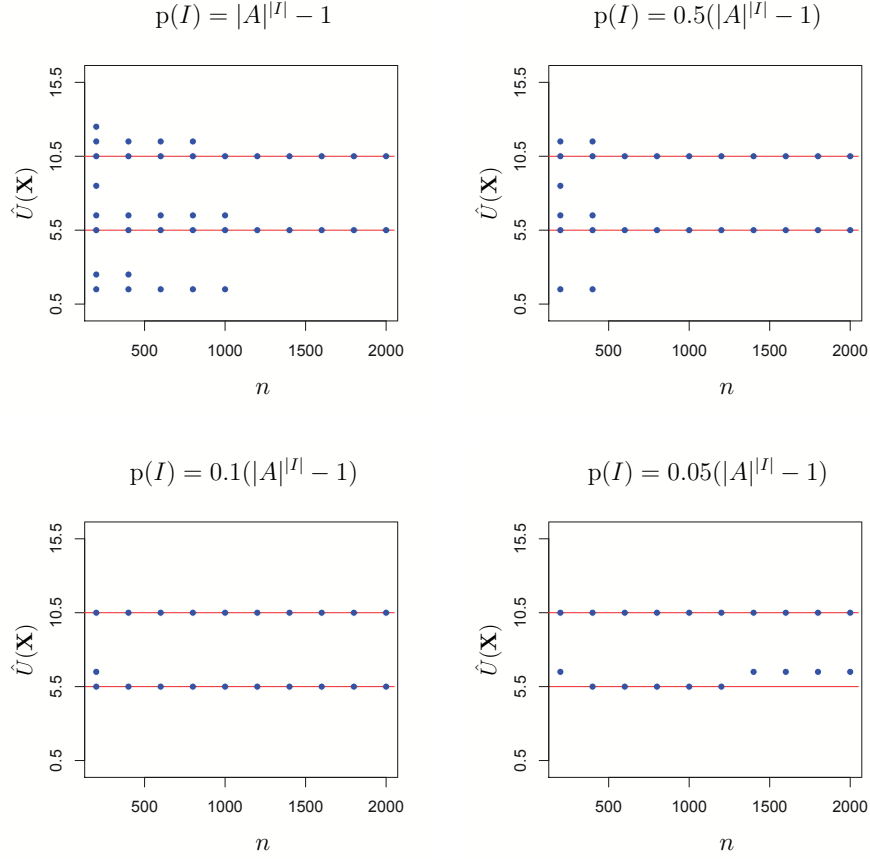


Figure 5: Hierarchical algorithm estimator of the set of points of independence $U^* = \{5.5, 10.5\}$, with $|A| = 2$ and for different penalizing functions.

We simulated n realizations of the vector $\vec{X} = (X_1, X_2, \dots, X_{15})$ such that X_1, \dots, X_5 follow a markov chain with transition matrix P_1 , X_6, \dots, X_{10} follow a markov chain with transition matrix P_2 and X_{11}, \dots, X_{15} follow again a markov chain with transition matrix P_1 . The initial condition for any one of these markov chains is the uniform distribution over A . The results for different values of n are given in Figure 4 for the exact estimator and in Figure 5 for the hierarchical algorithm.

We performed another simulation study for the alphabet $A = \{1, 2, 3\}$ and the transition matrices P'_1 and P'_2 given by

$$P'_1 = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{2}{3} & \frac{1}{3} \\ \frac{2}{3} & 0 & \frac{1}{3} \end{pmatrix}, \quad P'_2 = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{6} & \frac{1}{6} & 0 \end{pmatrix}.$$

We simulated as before n realizations of the vector $\vec{X} = (X_1, X_2, \dots, X_{15})$ such that X_1, \dots, X_5 follow a markov chain with transition matrix P'_1 , X_6, \dots, X_{10} follow a

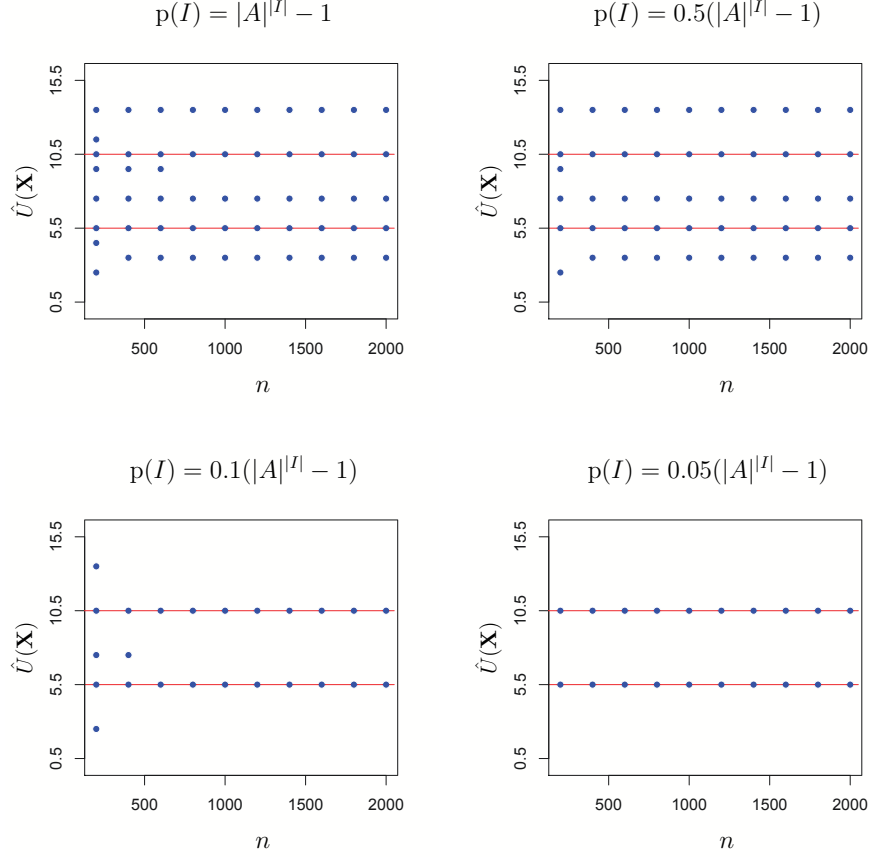


Figure 6: Dynamic programming estimator of the set of points of independence $U^* = \{5.5, 10.5\}$, with $|A| = 3$ for different penalizing functions.

markov chain with transition matrix P'_2 and X_{11}, \dots, X_{15} follow again a markov chain with transition matrix P'_1 . The initial condition for any one of these markov chains is the uniform distribution over A . The results for different values of n are given in Figure 6 for the exact estimator and in Figure 7 for the hierarchical algorithm.

The two algorithms were implemented in the open-source package R and are available upon request.

6 Dimensionality reduction in Ebola Virus protein alignments

As an illustration of the practical use of the method proposed in this paper we apply the dynamic programming algorithm described in Section 3 to a protein sequence alignment of Ebola Virus taken from the Ebola HFV Database [10].

The alignment corresponds to the matrix protein VP40, with $n = 21$ and $m = 326$. Although the possible amino-acids in each position are 20, the maximal observed

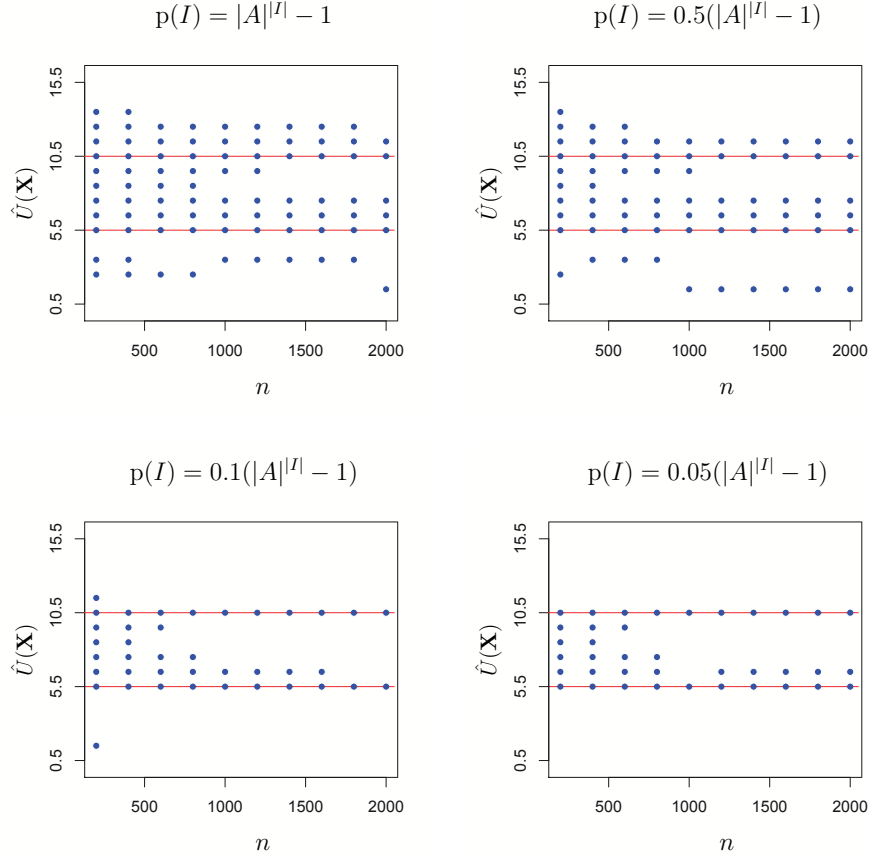


Figure 7: Hierarchical algorithm estimator of the set of points of independence $U^* = \{5.5, 10.5\}$, with $|A| = 3$ for different penalizing functions.

alphabet is A_{17} with 5 amino-acids. Moreover, although the number of columns is much bigger than the number of observations, a total of 206 columns in the alignment shows a conserved amino-acid, with no influence in the segmentation result.

As the number of observed amino-acids differs considerably in each position, we use the empirical penalizing function given by $\hat{p}(\mathbf{X}_I) = c(\max(2, \prod_{i \in I} \|\mathbf{X}_i\|_0) - 1)$, where $\|\mathbf{X}_i\|_0 = \sum_{a_i \in A_i} \mathbf{1}\{N(a_i) > 0\}$. In this case we obtained different segmentation patterns corresponding to different penalizing functions, that can be compared to the secondary structure patterns in [6] (see Figure 8).

Acknowledgments

F.L. is partially supported by a CNPq-Brazil fellowship (304836/2012-5) and FAPESP's fellowship (2014/00947-0). This article was produced as part of the activities of FAPESP Research, Innovation and Dissemination Center for Neuromathematics, grant

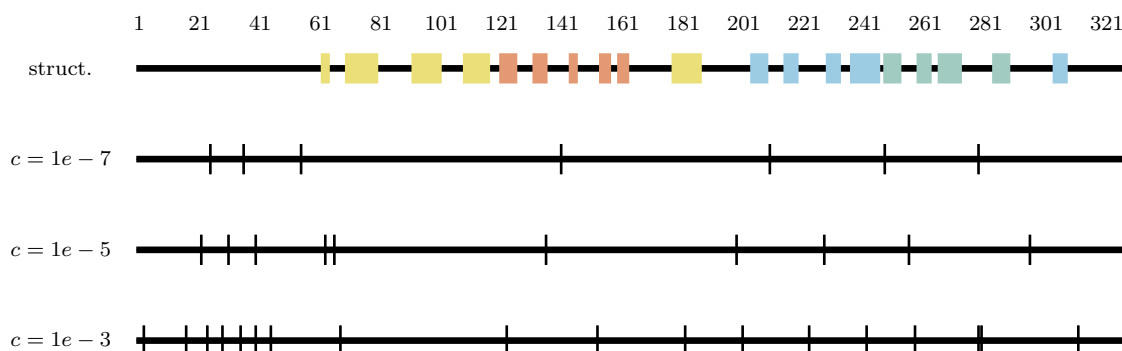


Figure 8: Sequence segmentation of Ebola virus protein VP40. The top represents the secondary structure of the protein described in [6], while the other lines represent the segments estimated by the dynamic programming algorithm for different values of c .

2013/07699-0, São Paulo Research Foundation.

References

- [1] J. Bai and P. Perron. Computation and analysis of multiple structural change models. *Journal of Applied Econometrics*, 18(1):1–22, 2003.
- [2] T.L. Bailey, M. Boden, F.A. Buske, M. Frith, C.E. Grant, L. Clementi, J. Ren, W.W. Li, and W.S. Noble. MEME Suite: tools for motif discovery and searching. *Nucleic Acids Research*, 37(suppl 2):W202–W208, 2009.
- [3] G. Bejerano, Y. Seldin, H. Margalit, and N. Tishby. Markovian domain fingerprinting: statistical segmentation of protein sequences. *Bioinformatics*, 17(10):927–934, 2001.
- [4] L. Breiman. *Probability*, volume 7 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992. Corrected reprint of the 1968 original.
- [5] I. Csiszar and Z. Talata. Context tree estimation for not necessarily finite memory processes, via bic and mdl. *Information Theory, IEEE Transactions on*, 52(3):1007–1016, March 2006.
- [6] A. Dessen, V. Volchkov, O. Dolnik, H.-D. Klenk, and W. Weissenhorn. Crystal structure of the matrix protein VP40 from Ebola virus. *The EMBO Journal*, 19(16):4228–4236, 2000.
- [7] P. Fryzlewicz. Wild binary segmentation for multiple change-point detection. *Ann. Statist.*, 42(6):2243–2281, 12 2014.

- [8] R. Gwadera, A. Gionis, and H. Mannila. Optimal segmentation using tree models. In *Data Mining, 2006. ICDM '06. Sixth International Conference on*, pages 244–253, Dec 2006.
- [9] D. M. Hawkins. Point estimation of the parameters of piecewise regression models. *J. Roy. Statist. Soc. Ser. C Appl. Statist.*, 25(1):51–57, 1976.
- [10] C. Kuiken, J. Thurmond, M. Dimitrijevic, and H. Yoon. The LANL hemorrhagic fever virus database, a new platform for analyzing biothreat viruses. *Nucleic Acids Res*, 40:D587-92, 2012.